AMENDMENT OF THE CLAIMS

The listing of claims below replaces all prior versions, and listings, of claims:

1   1.   (Original) A method of performing parallel data operations upon data in a
2   database, comprising:
3   receiving a data transaction request in a client system; and
4   executing a plurality of multi-phase parallel tasks in response to the
5   request to perform the data operations upon the data in the database.

1   2.   (Original) The method of claim 1, wherein receiving a data transaction
2   request comprises receiving a request for loading data into the database.

1   3.   (Original) The method of claim 1, wherein receiving a data transaction
2   request comprises receiving a request to perform a data transformation operation upon the
3   data in the database.

1   4.   (Original) The method of claim 3, wherein receiving a request to perform
2   the data transformation operation comprises receiving a request to perform one of a data
3   selection operation, a data validation operation, a data cleansing operation, and a data
4   query operation.

1   5.   (Currently Amended) The method of claim 1, wherein executing the
2   multi-phase parallel tasks comprises executing each of the parallel tasks in ~~one or more~~
3   <u>plural</u> phases.

1   6.   (Currently Amended) The method of claim 5, comprising executing a first
2   parallel task in a first number of phases and a second parallel task in a second<u>, different</u>
3   number of phases.

1   7.   (Currently Amended) The method of claim 5, further comprising each
2   parallel task providing a code to indicate if the task is to be re-invoked in the next phase<u>.</u>

1   8.   (Original) The method of claim 7, wherein providing the code comprises

2   providing the code to a task coordinator.

1   9.   (Original) The method of claim 8, wherein the code comprises a first code

2   to indicate that the task coordinator is to invoke a component in the next phase.

1   10.   (Original) The method of claim 8, wherein the code comprises a second

2   code to indicate that the task is not to invoke a component in the next phase.

1   11.   (Currently Amended) The method of claim 1, further comprising:

2   analyzing the transaction request;

3   creating a task plan in response to the transaction request;

4   implementing the task plan in a multi-phase organization;

5   executing a plurality of tasks in parallel to implement the task plan, in

6   response the launching of the task coordinator function;

7   determining whether an additional phase is required in order to execute the

8   task tasks; and

9   scheduling an additional phase in response to the determination that an

10   additional phase is required.

1   12.   (Original) The method of claim 11, wherein implementing the task plan

2   comprises creating a job script.

1   13.   (Currently Amended) The method of claim 11, wherein implementing the

2   task plan comprises:

3   translating the task plan;

4   selecting a plurality of software components to implement the translated

5   task plan;

6   assigning a plurality of processes corresponding to the software

7   components; and

8   creating a communications channel to allow for communications between

9   the processes.

1   14.   (Currently Amended) The method of claim 13, wherein selecting the

2   plurality of <u>software</u> components to implement the translated task plan comprises

3   selecting the plurality of <u>software</u> components to perform at least one of a data extraction

4   operation, a data transformation operation, and a data loading operation.

1   15.   (Currently Amended) An apparatus, comprising:

2   a user interface;

3   a processor coupled with the user interface, wherein the processor receives

4   a data transaction request from the user interface; and

5   a controller coupled with the processor, wherein the controller performs a

6   ~~number~~ <u>plurality</u> of tasks in parallel based upon instructions received from the processor,

7   each task performed in a plurality of phases.

1   16.   (Original) The apparatus of claim 15, wherein the processor generates a

2   task plan in response to the data transaction request.

1   17.   (Original) The apparatus of claim 16, wherein the controller comprises a

2   task coordinator to execute the task plan.

1   18.   (Original) The apparatus of claim 16, wherein the controller further

2   comprises a plurality of components to implement the task plan in parallel.

1   19.   (Original) The apparatus of claim 18, further comprising a

2   communications interface for enabling communications between the components.

1   20.   (Original) The apparatus of claim 18, wherein the controller further

2   comprises a storage unit for storing methods and functions to execute the task plan.

1   21.   (Currently Amended) The apparatus of claim 15, wherein the [[a]]

2 controller is coupled with the processor, wherein the controller performs a number of

3 tasks in parallel based upon instructions received from the processor, each task performed

4 in a plurality of phases further comprises the controller performing the tasks in a

5 sequence of multiple process steps.

1   22.   (Currently Amended) A system, comprising:

2   a database system; and

3   a network; and

4   a client system separate from the database system and coupled to the

5 database system over the network, the client system to establish plural sessions with the

6 database system to implement a plurality of data operations upon the database system in

7 parallel.

1   23.   (Cancelled)

1   24.   (Currently Amended) The system of claim 23 22, wherein the database

2 system is a parallel database system.

1   25.   (Currently Amended) The system of claim 22, wherein the client system

2 comprises:

3   a processor to receive a data transaction request;

4   a plurality of operators to perform parallel data operations in response to

5 the data transaction request;

6   an operator interface coupled to the operators, wherein the operator

7 interface allows communications between the operators.

1   26.   (Cancelled)

1   27.   (Cancelled)

1   28.   (Currently Amended) An article comprising at least one storage medium

2   containing instructions that when executed cause a client system to:

3   receive a data transaction request; and

4   execute a plurality of parallel tasks in response to the request to perform

5   data operations upon the data in ~~the~~ a database system over a network connection,

6   wherein the client system is separate from the database system.

1   29.   (Currently Amended) The article of claim 28, wherein the instructions

2   when executed cause the client system to execute each of the parallel tasks in ~~one or more~~

3   plural phases.

1   30.   (Currently Amended) The article of claim 29, wherein the instructions

2   when executed cause the client system to execute a first parallel task in a first number of

3   phases and a second parallel task in a second, different number of phases.

1   31.   (Original) The article of claim 29, wherein the instructions when executed

2   cause each parallel task to provide a code to indicate if the task is to be re-invoked in the

3   next phase.

1   32.   (Original) The article of claim 31, wherein the instructions when executed

2   cause the parallel task to provide the code to a task coordinator.

1   33.   (Original) The article of claim 32, wherein the code comprises a first code

2   to indicate that the task coordinator is to invoke a component in the next phase.

1   34.   (Original) The article of claim 32, wherein the code comprises a second

2   code to indicate that the task is not to invoke the component in the next phase.

1   35.   (Original) A method of performing parallel data operations upon data in a

2   database, comprising:

3   receiving a data transaction request; and

4   executing a plurality of synchronized multi-phase parallel tasks in

5   response to the request to perform the data operations upon the data in the database.

1   36.   (Currently Amended) The method of claim 35, wherein executing the

2   multi-phase parallel tasks comprises executing each of the parallel tasks in ~~one or more~~

3   <u>plural</u> phases.

1   37.   (Currently Amended) The method of claim 36, comprising executing a

2   first parallel task in a first number of phases and a second parallel task in a second<u>,</u>

3   <u>different</u> number of phases.

1   38.   (Original) The method of claim 36, further comprising each parallel task

2   providing a code to indicate if the task is to be re-invoked in the next phase.

1   39.   (Original) The method of claim 38, wherein providing the code comprises

2   providing the code to a task coordinator.

1   40.   (Original) The method of claim 39, wherein the code comprises a first

2   code to indicate that the task coordinator is to invoke a component in the next phase.

1   41.   (Original) The method of claim 39, wherein the code comprises a second

2   code to indicate that the task is not to invoke a component in the next phase.

1   42.   (Currently Amended) The method of claim 39, wherein the code

2   synchronizes the operation of one or more ~~component~~ <u>components</u>.
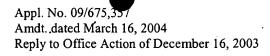
8

1   43.   (New) The method of claim 1, wherein executing the plurality of multi-

2   phase parallel tasks comprises:

3   executing at least first and second software components in parallel;

4   each of the first and second software components performing one or more

5   operations in a first phase;

6   waiting for a message from each of the first and second software

7   components prior to proceeding to a second phase; and

8   each of the first and second software components performing one or more

9   operations in the second phase.

1   44.   (New) The method of claim 43, further comprising:

2   waiting for another message from each of the first and second software

3   components prior to proceeding to a third phase;

4   the first software component performing one or more operations in the

5   third phase; and

6   the second software component being idle in the third phase.

1   45.   (New) The method of claim 44, further comprising:

2   receiving a first message from the first software component indicating that

3   the first software component is to be re-invoked in the third phase; and

4   receiving a second message from the second software component

5   indicating that the second component is not to be re-invoked in the third phase.

1   46.   (New) The apparatus of claim 15, wherein the controller comprises at least

2   first and second software components executable in parallel to perform the plurality of

3   tasks;

4   wherein each of the first and second software components is executable to

5   perform one or more operations in a first phase;

6   the controller to wait for a message from each of the first and second

7   software components prior to proceeding to a second phase; and

9

8   wherein each of the first and second software components is executable to

9   perform one or more operations in the second phase.

1   47.   (New) The apparatus of claim 46, wherein the controller is adapted to

2   further wait for another message from each of the first and second software components

3   prior to proceeding to a third phase;

4   wherein the first software component is executable to perform one or more

5   operations in the third phase, and the second software component is idle in the third

6   phase.

1   48.   (New) The apparatus of claim 47, wherein the controller is adapted to

2   further:

3   receive a first message from the first software component indicating that

4   the first software component is to be re-invoked in the third phase; and

5   receive a second message from the second software component indicating

6   that the second component is not to be re-invoked in the third phase.

1   49.   (New) The system of claim 22, wherein the client system is adapted to

2   execute plural tasks in parallel, each of the plural tasks executable in plural phases.

1   50.   (New) The article of claim 29, wherein executing each of the parallel tasks

2   in plural phases comprises:

3   executing at least first and second software components in parallel;

4   each of the first and second software components performing one or more

5   operations in a first phase;

6   waiting for a message from each of the first and second software

7   components prior to proceeding to a second phase; and

8   each of the first and second software components performing one or more

9   operations in the second phase.

1   51.   (New) The article of claim 51, wherein the instructions when executed

2 cause the client system to further:

3   wait for another message from each of the first and second software

4 components prior to proceeding to a third phase;

5   cause the first software component to perform one or more operations in

6 the third phase; and

7   cause the second software component to be idle in the third phase.

1   52.   (New) The article of claim 51, wherein the instructions when executed

2 cause the client system to further:

3   receive a first message from the first software component indicating that

4 the first software component is to be re-invoked in the third phase; and

5   receive a second message from the second software component indicating

6 that the second component is not to be re-invoked in the third phase.